Reproducing Kernel Hilbert Spaces (2)

Manel Martínez-Ramón

ECE, UNM

October, 2018

In the previous lesson we used a nonlinear transformation to pass from of $\mathbb{R}^2$ to $\mathbb{R}^p$, $p = \begin{pmatrix} 2+3 \\ 3 \end{pmatrix} = 10$:

$$1, x_1, x_2, x_1^2, x_2^2, x_1 x_2, x_1^2 x_2, x_1 x_2^2, x_1^3, x_2^3$$

In an input space of 2 dimensions, and with a Volterra expansion of order 5, we need 56 elements:

$$p = \begin{pmatrix} 2+5 \\ 5 \end{pmatrix} = 56$$

This is an example of the *the curse of dimensionality*, which we have to solve.

The Kernel trick

Trick to conjure up the curse of dimensionality:

- Find a method where we can work with expressions of *only the input space*. We have two fortunate facts:

1. If an algorithm fits the Representer Theorem, a dual expression can be constructed as function of dot products between data.
2. Functions that are dot products in higher dimension Hilbert Spaces exist.

The Kernel trick is nothing but the use of these two facts together.

Representer Theorem (Kimeldorf and Wahba, 1971)

- $\boldsymbol{\varphi}(\mathbf{x}_n) = \boldsymbol{\varphi}_n \in \mathcal{H}$ where $\mathcal{H}$ is a Hilbert space with dot product $\langle \boldsymbol{\varphi}_i, \boldsymbol{\varphi}_j \rangle) K(\mathbf{x}_i, \mathbf{x}_j)$
- $\Omega : [0, \infty) \rightarrow \mathbb{R}$ strictly monotonic increasing function
- $V : \left(\mathcal{X} \times \mathbb{R}^2\right)^N \rightarrow \mathbb{R} \cup \{\infty\}$ Arbitrary loss function

Then:

$$f^* = \min_{f \in \mathcal{H}} \left\{ V\left((f(\boldsymbol{\varphi}_1), \boldsymbol{\varphi}_1, y_1), \ldots, (f(\boldsymbol{\varphi}_N), \boldsymbol{\varphi}_N, y_N) \right) + \Omega(\|f\|_2^2) \right\}$$

admits a representation

$$f^*(\cdot) = \sum_{i=1}^{N} \alpha_i K(\cdot, \mathbf{x}_i), \quad \alpha_i \in \mathbb{R}, \ \ \boldsymbol{\alpha} \in \mathbb{R}^N$$

- Let us consider the Volterra case again.
- The estimator is

$$y[n] = \mathbf{w}^\top \boldsymbol{\varphi}(\mathbf{x}_n)$$

- and the MMSE solution as

$$\mathbf{w} = (\boldsymbol{\Phi}\boldsymbol{\Phi}^\top)^{-1}\boldsymbol{\Phi}\mathbf{y}$$

where $\boldsymbol{\Phi}$ is a matrix that contains all column vectors $\boldsymbol{\varphi}(\mathbf{x}_n)$.

Volterra revisited: dual

- Now, we take the fact that vector $\mathbf{w}$ is a linear function of the data as

$$\mathbf{w} = \sum_{n=1}^{N} \alpha_n \boldsymbol{\varphi}(\mathbf{x}_n) = \boldsymbol{\Phi}\boldsymbol{\alpha}$$

- We use this and previous equations together to obtain

$$\boldsymbol{\Phi}\boldsymbol{\alpha} = (\boldsymbol{\Phi}\boldsymbol{\Phi}^\top)^{-1}\boldsymbol{\Phi}\mathbf{y}$$

- By matrix manipulation we get the expression

$$\boldsymbol{\alpha} = (\boldsymbol{\Phi}^\top\boldsymbol{\Phi})^{-1}\mathbf{y}$$

- Here, matrix $\mathbf{K} = \boldsymbol{\Phi}^\top\boldsymbol{\Phi}$ contains all dot products between data.

- Also, since $\mathbf{w} = \boldsymbol{\Phi}\boldsymbol{\alpha}$ the estimator

$$y[m] = \mathbf{w}^\top \boldsymbol{\varphi}(\mathbf{x}_m)$$

becomes

$$y[m] = \boldsymbol{\alpha}^\top \boldsymbol{\Phi}^\top \boldsymbol{\varphi}(\mathbf{x}_m)$$

- This, in scalar notation is

$$y[m] = \sum_{n=1}^{N} \alpha_n < \boldsymbol{\varphi}(\mathbf{x}_n), \boldsymbol{\varphi}(\mathbf{x}_m) >$$

where $< \cdot, \cdot >$ denotes dot product between vectors.

Volterra revisited: kernel

- The next step would consist of finding a dot product in the higher dimension space that can be expressed as a function of the input space only.
- For the order 3 Volterra, this dot product is

$$< \boldsymbol{\varphi}(\mathbf{x}_n), \boldsymbol{\varphi}(\mathbf{x}_m) >= (\mathbf{x}_n^\top \mathbf{x}_m + 1)^3$$

- Hence, we have a compact representation that avoids the curse of dimensionality, since the term inside the parenthesis is just a scalar.

Volterra revisited:kernel

Let's prove it. Let $\mathbf{x}_1 = [x_1, x_2]^\top$ and $\mathbf{x}' = [x'_1, x'_2]^\top$ be two vectors:

$$\begin{aligned}(\mathbf{x}^\top\mathbf{x}' + 1)^3 &= (x_1x'_1 + x_2x'_2 + 1)^3 \\ &= x_1^3{x'}_1^3 + x_2^3{x'}_2^3 + 3x_1^2x_2{x'}_1^2{x'}_2 + 3x_1x_2^2{x'}_1{x'}_2^2 \\ &+ 3x_1^2{x'}_1^2 + 3x_2^2{x'}_2^2 + 6x_1x'_1x_2x'_2 + 3x_1x'_1 + 3x_2x'_2 + 1 =\end{aligned}$$

$(x_1x'_1 + x_2x'_2 + 1)^3 =$

$= [x_1^3,\ x_2^3,\ \sqrt{3}x_1^2x_2,\ \sqrt{3}x_1x_2^2,\ \sqrt{3}x_1^2,\ \sqrt(3)x_1^2,\ \sqrt{6}x_1x_2,\ \sqrt{3}x_1,\ \sqrt{3}x_2,\ 1]\cdot$
$[{x'}_1^3,\ {x'}_2^3,\ \sqrt{3}{x'}_1^2{x'}_2,\ \sqrt{3}{x'}_1{x'}_2^2,\ \sqrt{3}{x'}_1^2,\ \sqrt(3){x'}_1^2,\ \sqrt{6}{x'}_1{x'}_2,\ \sqrt{3}{x'}_1,$
$\sqrt{3}{x'}_2,\ 1]^\top$

Thus $(\mathbf{x}^\top\mathbf{x}' + 1)^3$ is the dot product of the Volterra expansion of the two vectors, up to some constants.

Outcomes of this lesson

- We have seen an example of a simple problem that cannot be solved using a linear classifier.
- A nonlinear estimator can be constructed by a nonlinear transformation to a space of higher dimension.
- This solution suffers from the curse of dimensionality.
- Nevertheless, using the Representer Theorem and finding a kernel dot product in this space, the problem is solved.